REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10 Jan 1997	Final

4. TITLE AND SUBTITLE
Case-Based Reasoning

5. FUNDING NUMBERS
N61339-96-D-0002

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lockheed Martin Corporation
Information Systems Company
12506 Lake Underhill Road
Orlando, FL 32825

8. PERFORMING ORGANIZATION REPORT NUMBER
ADST-II-CDRL-023R-9600242A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

NAWCTSD/STRICOM
12350 Research Parkway
Orlando, FL 32826-3224

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
A - Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

This research activity examines the current state-of-the-art in modeling the command decision process and implementing such models in software. The primary and initial target application is in automated command agents for DIS/ADS. This report was prepared for the Command Decision Modeling ADST II Delivery Order in accordance with the following documents:

a) Command Decision Modeling Overview (ADST-II-CDRL-023A-9600236)
b) Functional Description of a Command Agent (ADST-II-CDRL-023A-9600237)
c) Rule Based Systems (ADST-II-CDRL-023A-9600238)
d) Genetic Algorithms and Evolutionary Programming (ADST-II-CDRL-023A-9600239)
e) Petri Nets and Colored Petri Nets (ADST-II-CDRL-023A-9600240)
f) Neural Networks and Bounded Neural Networks (ADST-II-CDRL-023A-9600241)
g) Case-Based Reasoning (ADST-II-CDRL-023A-9600242)

14. SUBJECT TERMS
Case-Based; CBR; STRICOM; DIS; ADST-II;Simulation; DIS/ADS;

15. NUMBER OF PAGES
23

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

ADST-II-CDRL-023R-9600242A
10 January 97

ADVANCED DISTRIBUTED SIMULATION TECHNOLOGY II (ADST II)

DELIVERY ORDER # 0018

CDRL AB07

CASE-BASED REASONING

FOR: NAWCTSD/STRICOM
12350 Research Parkway
Orlando, FL 32826-3224
N61339-96-D-0002
DI-MISC-80711

BY: Lockheed Martin Corporation
Martin Marietta Technologies, Inc.
Information Systems Company
12506 Lake Underhill Road
Orlando, FL 32825

DOCUMENT CONTROL INFORMATION

Revision	Revision History	Date
	Original release	7/15/96
A	Incorporated customer comments	1/10/97
-		
-		
-		
-		
-		
-		
-		
-		
-		
-		
-		
-		

Table of Contents

PREFACE

This research activity examines the current state-of-the-art in modeling the command decision process and implementing such models in software. The primary and initial target application is in automated command agents for DIS/ADS. This report was prepared for the Command Decision Modeling ADST II Delivery Order in accordance with the following documents:

a) Command Decision Modeling Overview (ADST-II-CDRL-023A-9600236)
b) Functional Description of a Command Agent (ADST-II-CDRL-023A-9600237)
c) Rule Based Systems (ADST-II-CDRL-023A-9600238)
d) Genetic Algorithms and Evolutionary Programming (ADST-II-CDRL-023A-9600239)
e) Petri Nets and Colored Petri Nets (ADST-II-CDRL-023A-9600240)
f) Neural Networks and Bounded Neural Networks (ADST-II-CDRL-023A-9600241)
g) Case-Based Reasoning (ADST-II-CDRL-023A-9600242)

1.0 OVERVIEW OF TECHNOLOGY AREA

Many human problem solving techniques are based on reviewing previous problem experiences and applying their solutions to similar new problems [Gonzalez and Dankel, 1993]. This is the basis for Case-Based Reasoning (CBR) which uses a knowledge base of previous explicit experiences known as cases to solve problems. Similar cases are retrieved from a case-base and modified to fit the current situation. The new case is added to the case-base and each retrieved case is updated with the knowledge of whether it succeeded or failed in providing support for the solution of the new case. Unlike other reasoning paradigms such as rules, CBR is a machine learning technique that adapts to new situations and learns from them. How cases are adapted vary among implementations of CBR and are dependent on the domain of application.

One of the critical CBR tasks is the construction and organization of the case-base. Searching and reviewing of cases for applicability can be computationally expensive, so a proper set and organization of case indices must be developed that can be used to determine the similarity of cases. Indices usually consist of attributes that define the case, such as, inputs (the situation) and goals (solutions) it achieved. Proper refinement and weighting of these indices are of extreme importance not only to ensure that just relevant cases are retrieved, but also to reduce the total number of cases retrieved for review.

CBR is often compared against traditional rule-based reasoning. Both reasoning methodologies are patterned after human reasoning, but their knowledge representations are quite different. To represent knowledge as rules an expert must recollect and interpret the domain. CBR details the knowledge directly from historical cases and hence is a more formal and explicit representation of the domain by an expert [Gonzalez and Dankel, 1993]. Having a set of historical cases eases the knowledge acquisition process and allows the base of experience to come from more than one individual. If a library of historical cases does not exist, then creating the case-base can be more difficult than representing the knowledge as rules.

Since tactical decision-making has a great deal of historical examples, it is well-suited for CBR paradigm. An essential aspect of command reasoning is the need to plan courses of action. Case-Based Planning (CBP) is a form of CBR that is applicable to command reasoning systems. Cases consist of situational contexts and a solution plan to be executed. The situational context is composed of events that signify the situation (state of the world), any constraints, and the goals and sub-goals of the mission. The plan contains the actions necessary to achieve the goals in the case. CBP can be compared to both the state space representation and action ordering representation of generative planning [Blau et al., 1991]. The state space representation of previous plans is adapted using domain knowledge, current goals, and the context to obtain new plans.

The use of analogical reasoning is CBR's greatest advantage. New problems are solved by analogy with old ones and explanations are stated in terms of prior experience. The

power of CBR is drawn from a large case library and not from a set of first principles. To use these cases effectively, there must be a rich indexing mechanism. What the indices are and their order of importance depends on the domain. Thus, a general CBR system not only learns new cases as it proceeds, but also learns from experience a proper set of indices.

The primary disadvantage of the CBR involves issues associated with the computational cost of obtaining a solution. Searching through a case-base of poorly selected and indexed cases is expensive. A successful implementation of CBR depends on issues such as the quality of case indexing, case library architecture, case quality, and the proper number of cases. These issues are not inherent to the CBR technique but mere implementation problems to be overcome [Gonzalez and Dankel, 1993].

Case-based reasoning is applicable to a wide range of real-world situations. These situations can be knowledge rich, making construction of solutions complex. Cases can be built with specific knowledge about a problem, thereby reducing the effort to find the fewest relevant cases. Other real-world situations are weak in specific domain knowledge and only have access to domain-independent generally applicable knowledge for solving problems. Knowledge can be incomplete and evidence can be sparse. For these situations, CBR makes assumptions to fill in incomplete or missing knowledge based on cases in its case-base. The answers arrived at in this manner may not be optimal or not even completely correct, but CBR does provide a methodology to generate answers easily.

Cases provide a concrete starting point for deriving a solution. In these ways, CBR reduces the cognitive load involved in solving problems from the complex real-world [Kolodner, 1993].

2.0 NOTABLE IMPLEMENTATIONS AND VARIANTS

2.1 CBR and Computer Generated Forces

A recent use of CBR for modeling reactive behavior for Computer Generated Forces (CGF) focuses on the knowledge acquisition aspect of CBR research [Keirsey et al., 1995]. While the research does not explicitly use CBR for command and control or for command reasoning, the work does address the use of CBR for lower level behaviors. This work has the applicability of being scaled up into more complex upper echelon behaviors for tactical decision-making. The domain modeled is air combat. The choice of the air domain greatly simplifies the case structure since terrain does not need to be included. The technique does illustrate the capability of multiple autonomous agents to interact intelligently and effectively in both cooperative or competitive modes of operation.

Cases are defined simply using state variables that describe the motion and geometry of intelligent forces (IFOR) at a particular instant (e.g., angle off and range for air-combat domain). This describes the simulation space. Behavior responses are assigned to each case. The state variables form orthogonal axes of a multi-dimensional decision space. Thus, a case is defined as point in the decision space in terms of these axis state variables. Any specific configuration of vehicles in the simulation space can be mapped to a specific point in the decision space. A point in the decision space may map to a variety of configurations in the simulation space. Through careful selection of the variables that define the axes of the decision space, the simulation space may be designed to be invariant to rotation and translation of configurations in it.

Thus, a single point in the decision space maps to all possible rotations and translations of a particular configuration of vehicles. This helps reduce the number of applicable cases and aids in the case retrieval. The retrieved case corresponds to the point closest to the current situation point in the decision space. Actually, the decision space is divided into regions that represent different actions to be performed. These regions are defined by a Voronoi Diagram. A Voronoi Diagram for N cases in the decision space is a partitioning of the plane into N polygonal regions, one region associated with each case. Each point within a Voronoi region is closer to the case point for that region than it is to any other case point in the decision space [Kiersey et al., 1995].

For lower level behaviors, the retrieved case controls actions by turning on or off the appropriate behaviors (e.g., Pursue Threat and Avoid Threat) and selecting the relative priorities of component control laws. In many cases, control laws are set to an intermediate state giving them only partial influence over the ultimate control decision. Each control law is assigned a weight between zero and one. It is possible that this mechanism can be applied to command and control behavior, but more research in this area is required.

One of the problems with using the CBR approach here is that for complex multi-agent scenarios, the number of possible configurations of vehicles becomes too complex for the meaningful retrieval of cases. This necessitates the organization of an intelligent agent's perception against a more consistent and stable set of features [Kiersey et al., 1995]. In other words, a focus of selection is needed. For example, a focusing technique akin to human decision-making is focusing on a single target, real or hypothesized.

Another problem with this approach is that agents are unable to capture deep knowledge. This makes it difficult to indicate reasons why certain actions are performed and to incorporate this reasoning into the decision making process. Complex tactics are also difficult to capture with cases. Many times slightly different situations require drastically different actions. This leads to many cases that are similar making the retrieved case set large or making it difficult to provide enough cases to fully characterize the decision space. To reduce these difficulties a rich and effective case index mechanism must be devised.

2.2 Military History as a basis for CBR

An early association between CBR and tactical decision-making can be found in Dupuy [1988]. In this work, the deficiencies in traditional expert systems for command reasoning are highlighted while the appropriateness of CBR is suggested. Dupuy states traditional expert systems have not been very successful in the military domain due to several intractable problems [Dupuy, 1988]:

- the complexity of the interrelated and interdependent problems a commander must handle.
- the majority of military situations are dependent upon the individual, idiosyncratic behavior of large numbers of unpredictable human beings.
- the flow of engagements are never exactly the same as previous ones even if they look superficially similar.
- there are no living experts personally familiar will all the variety of combat situations that have occurred in the past.
- the continued addition of new weapons and other hardware make the current situations very different from past ones.
- the doctrine continually changes as weapons and equipment change.

Using military history as the basis for a case base can overcome these problems. Since there are patterns and similarities in the conduct and results of battles going back to 500

BC, a rich case-base can be constructed for military decision-making [Dupuy, 1988]. However, Dupuy only suggests an overall methodology without addressing the major issues or implementation. The problem of case matching to the current situation is mentioned but no substantial solutions are presented.

The emphasis of the work is on advising the commander more than autonomous command and control. This is quite common of the early CBR work which placed CBR in an advisory role [Dupuy, 1988; Wall et al., 1988; Goodman, 1989]. CBR advises the commander of possible course of actions (COAs), cautions, and alternatives with their associated risk. The approach is based upon the classical US Army approach to the estimate of a situation involving these steps [Dupuy, 1988]:

1) Assess the current situation based upon METT-T concerns. The results relate to indices in the historical case base.

2) Analysis of cases from historical combat. Indices are created for each case to facilitate categorization and comparison with the current situation. The details of a historical situation are elaborated, as is the extent to which action contributed to the success or failure of the case.

3) Review of expert's wisdom. A database containing the knowledge of past theorists and successful commanders who have written extensively on the art of war is used. The knowledge will be in the form of rules, hypotheses, laws, etc. that are related to the case indices and displayed to the commander.

4) Doctrine Review. The case indices are related to the current situation.

5) Analysis of Results. Identify what the winners did, losers did, the exceptions to do's and don'ts, and any special considerations or cautions.

6) Determination of possible courses of action, including the commander's and the enemy's.

7) Evaluation of courses of action. The feasibility of each COA is determined and success probabilities are assigned. The best COA is recommended along with errors to be avoided.

8) Decision. In this case the commander makes the decision but a command agent could easily choose the recommended COA.

This methodology serves only as a foundation for future CBR processes for tactical decision-making.

2.3 Case-Based Planning

Since tactical decision-making is based upon experience and involves uncertainty, CBR seems to be a natural fit. The primary application of CBR to command reasoning is in the area of case-based planning. In implementing case-based planning, three questions need to be addressed [Wall et al., 1988]:

1) How should case library be organized?
2) How should the retrieval be performed so that all relevant cases are retrieved?
3) How should the cases be presented so that a correct decision can be made?

The third question is a user interface issue requiring explanation and argumentation facilities that are of little use to an automated command reasoner. The first two questions, however, are very relevant to command decision modeling. Early work by Wall et al. [1988] suggests the use of dimensions as indices to organize cases. Dimensions have conditions that act as an initial filter and contain focus slots that identify what can be changed in a case to make the case stronger or weaker along a given dimension. It is the intersection of the situated cases with these dimensions that guides retrieval much like the CBR for CGF mentioned previously [Kiersey et al., 1995]. Dimensions allow the similarity of cases to be determined while still emphasizing the differences between the cases [Wall et al., 1988].

Scenario Generation. Case retrieval is guided by goal-directed reasoning. The focus is placed on the cases that are the most relevant to the success or failure of the given plan. This is especially important for military tactical planning in which many times the plans are predetermined (orders) but the actions to implement the plan are not specified. However, CBP can be taken one step further and determine the plan itself as well as the actions [Chaib-draa et al., 1993].

The CBP approach in this work centers around the generation of scenarios [Wall et al., 1988]. A scenario is simply defined as a collection of examples of possible future events. The information contained in the scenario is used to select a COA. The scenario depicts the events that might happen if a given COA is selected, the opponent follows an assumed response, and random processes, such as weather, act in assumed ways. This case base is divided up into categories depending upon these characteristics. The steps of scenario generation include [Wall et al., 1988]:

1) Describe current engagement and identify key situational factors including unit goals.

2) Given the possible COAs, potential future cases are retrieved.

3) Examples are used to explain the advantages or disadvantages of a given COA. Since this work considers CBP in an advisory role this is necessary.

4) The commander (or command agent in future systems) chooses COA based upon features of interest such as surprise or indirection.

5) The unit carries out the selected plan.

6) The events, decisions, and results are analyzed for the situational features that need to be modified or added to the case. The organization of the cases may also need to be modified for more efficient retrieval in the future.

The cases themselves are organized in a taxonomy based upon the type of engagement (1-on-1, 1-on-many, or overwhelmed), the force (friendly or enemy force), the components of the force (air support, strength of reserves, logistics, force-mobility, disposition, and command and control), the goal, and terrain factors (fields-of-fire, cover and concealment, mobility, key terrain, visibility, obstacles, and avenues-of-approach). These form the dimensions of the case. The information available to the case determines its granularity facilitating a multiple fidelity level approach. The network proximity based upon these dimensions determines the similarity and is used to generate hypothetical situations. Domain relationships are specified in the definition of the dimensions, not in the cases, and thus provides a general way to retrieve cases without depending upon explicit links in the cases themselves. Modifications are only made to the domain relationships. Cases themselves are not modified other than to modifying the case's dimensions. Using domain dimensions, cases become graph structure that can be matched based upon the taxonomy, implying a semantic similarity [Wall et al., 1988].

Each case contains:

- the start conditions of the situation.
- the sequence of events performed by the units and the corresponding results.
- the goals of the planner at the time.
- the intended actions, including the COA selected by the planner and the anticipated enemy actions.
- the disposition of opponent, i.e. the perceptions of the opponent's goals.
- any assumptions about domain unknowns or random processes (e.g., weather) and their effect, if any, on achieving the goals.
- the key aspects that led to success or failure of the engagement.

As a practical example of the CBP process, consider seven T-80 tank platoons advancing in echelon with heavily protected flanks and a headquarters (HQ) in the rear. A BLUFOR M-1 tank platoon had established contact with the OPFOR's left flank. Four BLUFOR tank platoons and a company HQ are located to the south of the engagement area. Due to

the presence of roads, both forces have MODERATE to HIGH mobility. Command and control attribute for the OPFOR is BELOW-AVERAGE since the HQ is out of visual range and because of the strict command hierarchy imposed on the OPFOR. Command and control attribute is GOOD for BLUFOR since they have an HQ present. The friendly force factors are classified as either MOBILE-FORCE, MODERATE-FORCE, or HIGHLY-MOBILE-FORCE. The enemy force factors are classified as either MODERATE-OFFENSIVE-FORCE or LARGE-OFFENSIVE-FORCE. The goal of the BLUFOR is DEFEND as ordered by the upper echelon commander. The cases retrieved were classified as MOBILE-DEFENSE, SCREENING-DEFENSE, MOBILE-OR-SCREENING-DEFENSE, COUNTER-ATTACK, and ASSUME-OBJECTIVE [Wall et al., 1988].

As an example of case, consider the MOBILE-DEFENSE case. It has the following representation [Wall et al., 1988]:

> <u>Case-ID</u>: MEETING-ENGAGEMENT
> Background: OPFOR units of unknown strength are exploiting a breach in the front lines. BLUFOR forces consist of an armored battalion with M-1's and two M-2 infantry companies. The objective is to delay and seal the breach if possible.
> <u>Course of Action</u>: MOBILE-DEFENSE
> <u>Results</u>: Heavy attrition occurred on both sides. OPFOR overextended itself and was exploited by a BLUFOR counterattack which was accomplished by splitting the force into wings.
> <u>Mobility factor</u>: Enemy force was reconnoitered quickly.
> <u>Deployment factor</u>: The BLUFOR forces were split opportunistically.

<u>Plan Advisor</u>. Another early work in CBP was done by Goodman [1989] for a CBR expert system that served as an advisor to a planner. As is with the work in Wall et al. [1988] there is no reason that with present technology it could not be adapted for an automated command reasoner. It focuses on the problem of military situational assessment and its role in planning; there is no complete military causal domain model. Much like the work of Dupuy [1988], a database of historical battles and prior planning information is needed. Also, plan modification needs to be supported [Goodman, 1989].

Cases are represented in the form of frames using the Land Warfare Database. Fifty-seven fields of data are divided into three groups: battle characteristics useful for case retrieval and prediction, predicted outcome information, and documentation/designation information. The frames use symbolic values to represent semantic information and contain links to related cases (same combatants, same attacker, same defender, etc.). Also included is a tactical map and historical commentary. The cases were developed at the constructive level of granularity. For example, the original Land Warfare Database contains twenty-seven distinct values for information on whether each side had an advantage or disadvantage concerning logistics [Goodman, 1989]. This was reduced to three values: attacker has the advantage, defender has the advantage, and neither has the advantage. Similar reductions were made for terrain, weather, etc. The symbols form

hierarchies that are used for case retrieval. For example, cases with low trafficability can be retrieved whether they are heavily wooded or consist of marshy terrain. For case base construction, the symbolic indices have the problem of fidelity when deciding to subclass them. For this work, classes were created only when a clear generalization between subclasses could be identified. Cases also consisted of formulas to derive new features, and like the symbolic indices, the granularity of the formulas is an issue.

The initial case is retrieved using nearest neighbor similarity function. Weights are assigned to the fields to be used in the matching. For symbolic fields, the inferential distance based upon the hierarchy is used for the weight. There are several shortcomings with this approach. From a knowledge engineering standpoint there can be problems determining the weighting factors. In this case, subject matter experts could come up with an initial weighting and could then refine the weight for different case retrievals. Over time, the experts could lose the context under which modifications were made causing previous cases to be retrieved incorrectly [Goodman, 1989]. In addition, it is difficult for a planner to determine what modifications need to be made to the plan in order to bring about a more favorable outcome.

This same problem also applies to autonomous planners. The problem can be reduced by presenting a list of similarities and differences in the input and by ordering retrieved cases by field weight. However, it is still difficult to interpret. Furthermore, the retrieval algorithm is very sensitive to small changes in the situation description which causes new cases to be retrieved. This is not easily understood by the planner and thus causes confusion. For an automated command reasoner, these presentation problems is not a problem since there is no symbolic to human thinking translation necessary. However, a methodology still needs to be developed to guide plan modification.

To counteract the previous problems, the nearest neighbor retrieval algorithm is replaced with an inductive discrimination analysis for knowledge acquisition. An interactive interface aids the subject matter experts in screening superfluous cases made during various consultations. [Goodman, 1989]. Traces of significant indices are displayed during a consultation to provide easier recognition of alternatives for plan modification.

When the planner is serving as an advisor the ease in which retrieved cases can be related back to the original situation is an important issue. An example the authors use is a commander defending a battle position in a prepared posture with the system returning similar cases where the defender won but used a fortified defense [Goodman, 1989]. This option may not be available to the commander in the field. The present solution is to introduce case prototypes which are conjunctions of indices which reflect conceptual categories of cases. For example, separate prototypes can be created for prepared and fortified defensive postures. Discrimination analysis only proceeds on cases indexed under each prototype. When a commander describes the current situation, the planning system returns only the cases where the defender is in a prepared defense. This situation demonstrates a possible problem with indices. The importance and usefulness of fields (indices) are context dependent and thus vary with the situation.

The research also noted that learning new failures automatically is difficult because subject matter experts must screen for causally meaningful indices. Indices must be automatically generated to explain these failures which is difficult to do. At the time this article was written the authors were investigating the addition of causal information to the system to guide automatic index generation [Goodman, 1989]. For an autonomous command reasoner this would be a necessity. Without it , the reasoner could never learn from its mistakes or even its successes.

<u>Military Transportation Planning</u>. A more recent work in CBP was performed in the domain of military transportation planning and scheduling [Blau et al., 1991]. While this area is not the same as that of autonomous command agents, it is related to command and control and does illustrate important aspects of CBP. For example, the research focuses on domains where multiple agents, with different goals and viewpoints, attempt to plan strategies using incomplete, or uncertain information. The domain of tactical decision-making fits under this definition. In addition, cases in these domains develop over time, requiring reasoning about sequences of events. Military Transportation Planning and Scheduling is somewhat simpler in that all agents involved are under at least partial control of a central agent, and are all working cooperatively. However, the actions in this domain fail frequently due to equipment failures, changing environment conditions, etc. This form of logistics is especially concerned with trade-offs to maximize goal attainment which has direct applicability to tactical decision-making.

Most CBP systems deal with cases consisting of a plan and information about its execution. Blau et al. [1991] focus their attention on the development of a case representation language (CRL) for these domains. The CRL provides for the dynamic aspects of these cases. It is designed to represent cases consisting of the top-level goal(s) and information about states and events [Blau et al., 1991]. The CRL provides for two types of knowledge: general domain knowledge and cases. The general domain knowledge consists of objects, actions, goals, and plans in the domain. The cases consist of situation/solution pairs where the situation consists of the top-level goal(s) and a starting state. The solution consists of the representation of the observable portion of the agent's execution of the plan necessary to satisfy the goals (a network of events and linear sequence of states). The cases are built from instances of the classes in the domain knowledge hierarchies.

Domain knowledge is organized into classification hierarchies, linked to one another via frame slots. Example domain hierarchies include [Blau et al., 1991]:

Object Hierarchy:
    C-130 IS A cargo plane
    capacity: 42673 lbs
    range: 2356 miles

Action Hierarchy:

TAKEOFF IS A DEPARTURE
vehicle: :type AIR-VEHICLE
from-location: :type AIRSTRIP
destination: :type AIRSTRIP
actor:
procedure: (takeoff vehicle from-location)

The goal/plan hierarchy contains an And/Or tree for goal/plan decomposition. A Goal is defined as an OR Node whose inputs are links to alternative plans [Blau et al., 1991].
A plan is defined as an AND Node with links to all required sub-goals. The hierarchy is incomplete because it represents the known partial knowledge of the domain which is why CBP is necessary in the first place. Added cases can also add to this domain knowledge.

Instances of domain objects fill in the slots of frames in these hierarchies. These objects (frames) also contain Part-Of links to connect actions to interpretations and expectation links that provide some partial causal model of the domain. Interpretations are explanations for the occurrence of set of events with a degree of certainty. They may contain some local knowledge such as a goal, objects affected by the interpretation, etc.
Interpretation links can be implicit or explicit. Implicit links are those in which other actions are expected as part of the same interpretation. Explicit links are concerned with different interpretations or actions that are not in the same interpretation, e.g. defensive actions expected in response to an offensive action.

Cases are formed by creating links between the various hierarchies. Event and states are related by causal, temporal, and membership links. Also included are behaviors that can cause a state change and events that the state enables. Goals specify values that a state should hold, i.e. a goal state. For goals that contain sub-goals, fuzzy values are used to represent goal attainment. Events associate action and state information by joining an action with the goal achieved and with the information about the execution of the action. Events contain goals, actions, costs (resources), contexts, and time intervals. They are linked causally to both state changes and other events with a measure of certainty (fuzzy). Events can also be linked to a state which enabled the event. For example, the following would be the representation of a cargo transfer event [Blau et al., 1991]:

TRANSFER-CARGO-001 IS A TRANSFER-CARGO
from: WAREHOUSE-001
to: WESTOVER
cargo: (contents WAREHOUSE-001)
goal: '(((location cargo) to))
parts: '(LOAD-001 TAKEOFF-001 LAND-234 UNLOAD-23)

Cases are retrieved based upon the description of the situation and the case's similarity to the situation based upon relevant and salient features. The most salient features for initial retrieval are its top level goals and available resources. The similarity is determined

based upon the inferential distance of salient features from abstractions in hierarchies. Currently only a strict metric is being used that prefers similarity to parents and grandparents instead of siblings.

Once a case is retrieved, plan adaptation is performed on the parts of the solution which are not applicable or repeatable, because of a lack of resources. These parts are individually adapted by using substitution (replacing a sub-plan for a sibling node in a goal-plan taxonomy), compression's (e.g., eliminating the step from the plan and substituting dependent sub-plans), extensions (e.g., generalizing another sub-plan to cover and replace current one), and recursively performing CBR on sub-plans (which may have been achieved or modified previously) [Blau et al., 1991].

In many cases the plan can be simulated or assessed by other analytical means in order to predict the success or failure of the plan (see Lee [1995]), to find any tradeoffs, to identify possible sources of failures, and to determine the cost associated with the plan. For the military transportation domain, scheduling algorithms and analytical techniques are used [Blau et al, 1991]. Finally, the tradeoffs are accepted or plan is re-modified under a different adaptation rule, or another plan must be selected for adaptation if the retrieved plan cannot be modified.

This work has promise for command reasoning because of its sound implementation foundation. It also deals with many of the same problems faced in tactical decision-making, specifically goal attainment and tradeoffs. No future work as yet to emerge.

<u>RAND Integrated Simulation Environment</u>. CBP was actually implemented in the extension of the RAND Integrated Simulation Environment (RISE) [Catsimpoolas and Marti, 1992]. For RISE, higher echelon behaviors were implemented using scripts describing goal-directed behavior in terms of lower level scripts and so on down to the primitive level. Lower level scripts are created by first analyzing the top level plan and passing the appropriate portions to the lower levels where more planning is initiated. This is done recursively down to the platform level where the behaviors become primitive.

The basis of the RISE work is situational assessment to support planning. The state information and world view of each agent is used to select and implement plans. The scripts are the instantiation of a plan which, in the dynamic military domain, are expected to fail frequently [Catsimpoolas and Marti, 1992]. Thus, the scripts contain contingencies for retrying steps or trying alternate paths instead of costly replanning. However, replanning is performed if necessary. The scripts encapsulate military knowledge and are matched against the current situation. One a similar match is found, they may be modified and are either executed or distributed to subordinate units.

A plan consists of a series of scripts that define the events to be executed. Each step during the plan is composed of goals that may be successful, may fail, or need to be retried. If the successful step was the last of the subplans then the higher level plan also

completes. If the subplan failed then the remainder of the subplan is discarded and the higher level reasoners either replan the step or signal the failure up the chain of command. The goal graph built during recursive decomposition can be traversed differently depending upon which goals are failures and which are successful. If necessary the current script is considered a failure and replanning occurs. Replanning occurs when a lower level goal fails but a higher level goal remains valid. When necessary, the lower level goals are recursively replanned up to the highest level if necessary. The script itself is similar to programming statements that are events that mark the passage of simulation time. For example, they may invoke other scripts, perform conditionals, and perform loops.

The scripts are labeled so that the case-based planner can modify the script to conform to the current situation. Portions of the script can be removed or added as needed. The selected and modified scripts become the orders given to the units. They may invoke further planning all the way down to the lowest levels where the scripts contain only primitive actions and planning activities. A small initial case pool is used initially and thus many generic scripts are modified during execution to perform activities. This is a tradeoff between the space and time required to support the enormous number of cases necessary to produce exact matches versus the time required to generate scripts from scratch with no case-base at all.

The situational assessment aspect of the planner concerns the situational geometry of the situation. The situation is defined in terms of a partial ordering among objects of interest in the local environment. The actions include [Catsimpoolas and Marti, 1992]:

- locating the objects of interest.
- rotating all the objects to a coordinate system based upon the line from the object to its objective.
- computing the partial orderings to satisfy the left of, right of, in front of, and behind relations.
- computing the near, far, between, and line-of-sight relations, and a situation description data structure.

The planning involves subordinate objects, their assigned tasks, and the situation geometry. The current situation is not only defined by the situation geometry but also any cooperating units, the object state, future objectives, etc. This description is used to match against the case base. The script associated with the selected case is then modified by the planner and assigned to the object in question.

The RISE CBR method was tested on simple, aggregate level BLUFOR versus OPFOR exercises in which a BLUFOR is trying to reach a destination that is guarded by an OPFOR. The case base initially only contains case scripts that instruct the units how to

fight the enemy, fight for a location objective, and fight a nearby enemy who is not blocking the unit. The results look promising but the experiments are very simple when compared to the tactical decision-making required of upper echelon units in a virtual simulation. Their future work includes expanding the granularity of the case base and more efficient case searching and matching [Catsimpoolas and Marti, 1992].

2.4 RPD Architecture

An early implementation of case-base reasoning for command-level tactical decision-making is the RPD architecture [Chaib-draa et al.; 1993]. It is a hybrid architecture design to facilitate the coordination of intelligent agents to promote beneficial interactions and avoid harmful ones. Since local decisions may have global impacts, this becomes an important issue. The architecture is composed of three primary components: a reactive component, a planning component, and a deliberative component, hence the name RPD. The deliberative component handles decision making in completely unfamiliar environments, e.g. when planning fails, and takes into account intentions of others.

Under the RPD architecture, information from the environment is perceived and either a reactive action occurs or a planning action occurs if the information is in the form of a goal. If the information is not perceived as an action or a goal then identification and recognition are needed. If the information is ambiguous or a goal needs elaboration due an unfamiliar environment then deliberative decision making is performed in order to commit to achieving a goal that spawns the planning.

The identification system uses case-based reasoning for situations that the rule based planner cannot identify. It works like an indexing system that uses the perceived information and similarity metrics to discriminate the goal and action information in the memory. The hierarchy between situations is then used to decide which goal or action must be executed. For ambiguous goals or actions, the deliberative component is used.

Using CBP, a command agent must anticipate problems in order to find in memory a plan that avoids those problems. It must then extract this plan and adapt the plan if necessary. If the plan has a flaw then it must repair it by looking for causal explanations of the flaw in order to determine the failure(s). Next, it must determine the strategy necessary to arrange the plan when the cause of the flaw is found. Finally, any successful plans are saved to memory indexed by the goals they satisfy and the problems they avoid.

The identification and recognition module and the planning module for the RPD architecture both use CBR. Both have a memory hierarchy where general cases are represented at the top and concrete cases at the bottom. The identification and recognition module use this memory to identify the current situation and generate the goal that best suits the situation. This can be done in two ways: If the situation is familiar, the recognition sub-module identifies the proper goal using rules that can be dynamically

modified. For unfamiliar situations the identification sub-module uses case-based identification. In order to retrieve goals, the memory hierarchy is organized with situations and their corresponding goals. Of course similarity metrics must be determined to compare the situations, but actually, this problem is very specific to each application domain and the authors have not considered any general solution.

The CBP uses the mental state of the agent (beliefs, intentions, and commitments), which are continually modified, to modify its plans. First a plan that was used to accomplish a similar goal under a similar mental state is found. The memory hierarchy links previous mental states of the agent to the plans used in those circumstances. Similarity metrics are also needed to compare the mental states. Once a plan is found, it needs to be modified if it does not exactly meet the requirements of the current mental state.

The repair process is an essential mechanism because the plan may have flaws or be the "wrong" case (i.e., goals not do not apply may be identified for the given situation). This aspect was not investigated by the research. For an intelligent command agent that can continually adapt and learn, case repair is an important process. CBR alone will not solve all the command reasoning problems. CBR and CBP will play a crucial role in the hybrid approaches that will be necessary for autonomous command agents. Unfortunately, the RPD architecture does not provide any implemented evidence to support the CBR role in agent planning.

2.5 Intelligent Agents

CBR has often been used under the context of intelligent agents. Simulating intelligent agents with CBR addresses several command modeling issues, namely anticipation, experiential learning, failure avoidance, goal attainment, and adaptive behavior [Castillo, 1991]. Intelligent agents possess the ability to anticipate, learn, and perceive situations about the real world. In the research by Castillo [1991], an intelligent agent is an object containing rule sets, plans, constraints (resources and requirements of the plan), methods, and a cognitive system. The CBR planner is one aspect of the cognitive system. The cognitive system also contains a sensory filter, problem anticipator, and a rule base containing prediction rules, operating rules, utilization rules, modification rules, and similarity metric rules. Rules in this context are used to support the CBR process and not to provide intelligent behavior on their own.

This work emphasizes specially designed cases to support goal directed reasoning, anticipation, dynamic memory, plan organization, and retrieval [Castillo, 1991]. The intelligent agents contain models of their decision making components and use them when faced with alternatives. They formulate plans and alter plans according to the demands of their environment. As shown in previous work, planning is the most important aspect of an intelligent agent's cognitive activity and provides a mechanism for dealing with a dynamic environment [Wall, 1988; Goodman, 1989; Blau et al., 1991; Chaib-draa et al, 1993; Catsimpoolas, 1992; Castillo, 1991].

Cases are used in conjunction with scripts and rules to provide intelligent behavior. Unlike traditional case-based reasoning, a heterogeneous case base is used containing different formal representations of plans, requests, goals, etc. Each case is represented as an aggregation of these objects such as the goals they satisfy and the problems they avoid. Thus, the case-base is not made up of cases per se, but it is a discrimination network with multiple entry and exit points. The network has with three types of interconnecting semantic relations: logical (specializations/subclasses), structural (aggregation/decomposition of objects), and causal (plans and goals that satisfy or do not satisfy the plan). Any or all of these relations can be used to derive cases. These relations connect the different homogeneous layers of the case base and traditional CBR techniques can be performed for intra-level organization.

The scripts themselves are similar to frames except they represent dynamic behavior in a declarative fashion. They are dynamically altered and hierarchically composed. These are the solution plans that are represented in the case base. They are abstractions of atomic activities that define the actual events comprising the agent's plan. Thus, scripts are the realizations of plans. Plans specify the goals or intentions of the agent. This sequence includes temporal considerations, resource constraints, and trade-off opportunities. The goals are also prioritized so partial achievement of the goals is not necessarily considered a failure. The CBR's role is to match actions that satisfy similar goals. The goals themselves can also have varying fidelity (general to specific) and degrees of decomposition (subgoals). Goals are frames containing semantic links, procedural attachments, and production rules stored in the form of an And/Or graph.

The intentions of the intelligent agent are the driving force behind case selection and are provided in the form of a request. The plans and corresponding scripts chosen are those most likely to achieve the objectives of the request. Case retrieval involves the use of similarity metrics to compare goals, plans, etc. Three metrics are used: a taxonomic similarity metric that measures the closeness in the specialization hierarchy, a feature similarity metric that compares attributes and their values, and an importance metric. Special heuristics are used to compute these metrics for each kind of knowledge unit in the case base. However, before these metrics are applied, an anticipation step is first performed in which failure avoidance is the focus. Failure cases are indexed by the goals they failed to accomplish and contain all the plans, goals, and scripts that were involved in the failure. Any previous attempts to satisfy the same or similar goals are examined to see if those attempts were a success or not. The anticipator attempts to predict future event behavior using domain independent heuristics designed to predict the state of a location or resource at some future point in time. The retrieval step follows. If no exact matches are found the case base is searched for scripts that address the goals specified in the request and a new case is built. Goals can also be incrementally removed until a match is found and then merged with scripts that satisfy the removed goals. If this is not successful, then the system searches for partial matches. In this situation, different requests, goals, and scripts are substituted in order to find a similar plan in the case-base. When CBR cannot determine a plan, it resorts to learning by inquiry from other agents in

the system [Castillo, 1991]. This is an area of further research and suggests, as shown previously, a hybrid architecture.

Once the plan is retrieved it may need to be modified. This can take on several forms: modifying activity durations, removing activities, adding activities, replacing activities, modifying temporal relations, or formulating new behavior. Next, the case is executed and assessed. During plan assessment, new cases are added to the case base as failures or successes. The repair of failures is delegated back to the plan modifier. Performance criteria are used to determine success. They include: total script duration, total resource time, number of unique locations, number of unique resources, number of preemptions, number of precondition violations, number of run-time failures, and percentage of goals satisfied [Castillo, 1991].

A common problem in CBR is transforming the data in the environment into a form that can be used to select cases. The environment, especially the military environment, contains infinite variations in, for example, equipment, positions, and weather, for the same situation. These variations that must be coalesced into a single cohesive case representation. This work removes the infinite variation problem by assuming all the cases and the inputs are presented in terms of abstract goals written in the form of a script. This is so the research could better focus on the command and control aspect of the problem rather than the environment translation.

This work was not applied to a real time system so its performance in a command and control environment is unclear. The partial matches were time consuming. This is an aspect of case-based reasoning in general, but it may affect realistic command agent performance. The approach does set the foundation of the architecture necessary for command and control. It should be noted that no one paradigm will be sufficient for command reasoning. A hybrid scheme combining the best of the paradigms mentioned in this report that address the specific segments of cognitive processing will be necessary. Finally, the number of knowledge units needed to perform basic level agent behavior was too large and, therefore, impractical [Castillo, 1991]. The similarity metrics alone required an incredible amount of knowledge. However, these learning agents are a step in the right direction for realistic command agents.

The Judgmental METT-T work mentioned previously used to concept of multiple intelligent agents working together to provide the tactical decision-making of a command agent [Mall et al., 1995]. Agents handle command reasoning aspects such as the mission and enemy concerns. The CBR aspect of this command reasoner was concerned with using CBR to select course of actions when problem situations arose.

Problem situations are generated by other intelligent agents in the system in response to situations such as a bridge out or enemy presence. The alternative generator intelligent agent uses a reduced form of CBR to retrieve solutions from a case base of problem solutions. Problem situations are characterized by their class, type, and origin. Using rules, the solutions are matched first by the problem class, returning general solutions and

then by the type that specify specific solutions of the general solution. The agent further discriminates alternate possibilities since some solutions require a specific echelon level in order to be implemented. For example, the problem situation of a tank platoon facing a bridge out on its route would return general solutions such as reroute or breach-obstacle from the case base (since its class is an obstacle). Since the problem is of type bridge-out, more specific alternatives such as find-river-crossing or build-temporary-bridge are retrieved. Since the unit involved is a tank platoon and a bridge platoon is necessary for a bridge, this alternative cannot be implemented. The resources and time required to implement the alternative are also used to eliminate possible alternatives. This work requires exact matches of problem to alternative and will use general alternatives when specific ones do not exist. No attempt to modify cases and store them for future use, i.e. learn, was done. This serves as simply as a portion of a more complex CBR command agent.

3.0 APPLICABILITY TO COMMAND DECISION MODELING

Tactical decision-making is based upon experience and involves uncertainty. Since there are patterns and similarities in the conduct and results of battles going back to 500 BC, a case-based approach seems applicable. However, the lack of robust implementations does not support (or discredit for that matter) its applicability. Like the work in Wall et al. [1988] there is no reason that with present technology it could not be adapted for an automated command reasoner.

Some problems to be faced by CBR for command agents include:

- transforming the environment into a situation description that facilitates the meaningful retrieval of cases. The infinite varieties of the environment make this a difficult task.

- selecting proper indices and indices for failures to facilitate learning from mistakes.

- case base organization. How the case base is decomposed, indices applied, and case links affect the case retrieval efficiency and correctness.

- being unable to capture deep knowledge. This makes it impossible to indicate reasons why certain actions are performed and to incorporate this reasoning into the decision making process.

- difficulty in capturing complex tactics with cases. In some instances, when slightly different situations required drastically different actions, there could be difficulty in providing enough cases to fully characterize the domain. Difficulties with case mismatching must be resolved through the careful selection of the features used in matching.

As mentioned previously, a hybrid architecture is probably required to model all the aspects of a command agent. In whatever architecture is selected, CBR will most certainly play a role. The ability of CBR to match against previous similar situations to provide actions or plans and modify these results and store them for later use is a requirement for command decision modeling.

4.0 CONCLUSIONS

[Optional. Not Required.]

5.0 REFERENCES

Blau, Lauren; **Bonissone**, Piero P.; **Ayub**, Saad, "Planning with Dynamic Cases", *Proceedings of a Workshop on Case-Based Reasoning*, Washington D. C., May 8-10 1991, pp. 295-306.

Castillo, David, *Toward a Paradigm for Simulation Intelligent Agents*, Doctoral Dissertation, University of Central Florida, 1991.

Catsimpoolas, Niels; **Marti**, Jed, "Scripting Highly Autonomous Simulation Behavior Using Case-Based Reasoning", *Proceedings of the 25th Annual Simulation Symposium*, University of Florida, DARPA, Army Research Office, Gainesville, FL, December 7-9 1994, pp. 13-19.

Chaib-draa, B.; **Paquet**, E.; **Lamontagne**, L., "Integrating Reaction, Planning, and Deliberation in Architecture for Multiagent Environment", *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation,* Institute for Simulation and Training, Orlando, FL, March 17-19 1993, pp. 45-55.

Dupuy, T. N., "Military History and Case-Based Reasoning", *Proceedings of a Workshop on Case-Based Reasoning*, Clearwater Beach, Florida, May 10-13, 1988, pp. 125-135

Gonzalez, Avelino J.; **Dankel**, Douglas D., *The Engineering of Knowledge-Based Systems Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1993.

Goodman, Marc, "CBR in Battle Planning", *Proceedings of a Workshop on Case-Based Reasoning*, Morgan Kaufmann: San Mateo, CA, Pensacola Beach, Florida, May 31- June 2 1989, pp. 264-269.

Kiersey, David; **Krozel**, Jimmy; **Payton**, David; **Tseng**, David, "Case-Based Computer Generated Forces", *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation,* Institute for Simulation and Training, Orlando, FL, May 9-11 1995, pp. 307-315.

Kolodner, Janet, *Case-Base Reasoning,* Morgan Kaufmann Publishers, Inc. San Mateo, CA, 1993.

Lee, Jin Joo; **Fishwick**, Paul A., "Simulation-Based Planning for Computer Generated Forces", *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, Institute for Simulation and Training, Orlando, FL, May 4-6 1994, pp. 220-235.

Mall, Howard; **Bimson**, Kent; **McCormack**, Jenifer; **Ourston**, Dirk, "Command Entity Cognitive Behaviors for SAF and CGF", *Proceedings of the Fifth Conference on*

Computer Generated Forces and Behavioral Representation, Institute for Simulation and Training, Orlando, FL, May 9-11 1995, pp. 203-208.

Wall, Rajendra S.; **Donahue**, Dan; **Hill**, Stan, "The Use of Domain Semantics for Retrieval and Explanation in Case-Based Reasoning", *Proceedings of the 1988 DARPA CBR Workshop*, Morgan Kaufmann: San Mateo, CA, Clearwater Beach, FL, May 10-13 1988, pp. 447-462.